NIMbox/NEMbox
FPGA / VHDL
Self Made Firmware Handbook

User's Manual

*0000.A0

General Remarks

The only purpose of this manual is a description of the product. It must not be interpreted as a declaration of conformity for this product including the product and software.

W-Ie-Ne-R revises this product and manual without notice. Differences between the description in manual and the product are possible.

W-Ie-Ne-R excludes completely any liability for loss of profits, loss of business, loss of use or data, interrupt of business, or for indirect, special incidental, or consequential damages of any kind, even if **W-Ie-Ne-R** has been advises of the possibility of such damages arising from any defect or error in this manual or product.

Any use of the product which may influence health of human beings requires the express written permission of **W-Ie-Ne-R**.

Products mentioned in this manual are mentioned for identification purposes only. Product names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies.

No part of this product, including the product and the software may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the express written permission of **W-Ie-Ne-R**.

Table of contents:

1 Advanced usage through installing self made firmware 3

1.1 Foreword 3

1.2 Important Comments 3

2 Download of Firmware Files, JTAG and USB Interface 4

2.1 JTAG Interface Download 4

2.2 WINDOWS Driver for USB FX2 5

2.3 Firmware download via USB 5

3 Hardware description 6

3.1 General description of NIMbox/NEMbox 6

3.2 Hardware configuration 6

3.3 FPGA I/O Pin Assignment List UCF File 7

3.4 FX2 USB I/O, RESET, LED and CLOCK 10

3.5 I/O Connectors 10

3.6 I/O submodules 11

3.7 SU703 – Discriminator I/O 12

3.7.1 General Description 12

3.7.2 SU703 (discriminator) pin assignment 12

3.8 SU704 – NIM I/O 13

3.8.1 General Description 13

3.8.2 SU704 (NIM I/O) pin assignment 13

3.9 SU706 – Fast ADC submodule 14

3.9.1 General Description 14

3.9.2 SU706 (Fast ADC) pin assignment 14

3.10 SU707 – LVDS I/O submodule 15

3.10.1 General Description 15

3.10.2 SU707 (LVDS IO) pin assignment 15

3.11 SU709 – Temperature Sensors submodule 16

3.11.1 General Description 16

3.11.2 SU709 (Temperature Sensors) pin assignment 16

3.12 SU710 – Fast DAC submodule 17

3.12.1 General Description 17

3.12.2 SU710 (Fast DAC) pin assignment 17

3.13 SU711 – 5 Channel Delay 0.5 to 128 ns 18

3.13.1 General Description 18

3.13.2 SU710 (Fast DAC) pin assignment 18

1 Advanced usage through installing self made firmware

1.1 Foreword

The NIMbox/NEMbox is normally shipped with preinstalled firmware which is consistent with the hardware version and with the position of the installed submodules on the main FPGA board. With four positions and a handful of submodule types there exist statistically too many variants for firmware and front panels that can all be supported by W-Ie-Ne-R.

Hence up to a dozen of decisive NIMbox/NEMbox selections can be ordered from stock while individual custom solutions can be offered on special request.

For these selections Wiener Plein & Baus can offer the full LabView™-support with also adaptive firmware selections of hardware useable LabView™ library symbols.

Intrinsically the NIMbox/NEMbox concept allows for even more individual or "home brewed" solutions whether they concern mixing of W-Ie-Ne-R submodules or even self made ones when generating the own FPGA firmware. Hence within a portable NEMbox case or as a NIMbox module in a Mini-Crate one can easily prepare an individual kind of portable Mini-Laboratory. Mechanics, connectors and form factors are made simple and have the potential to realize own ideas with much reduced effort.

Additional advantage of self programming firmware is that no FPGA resources may be wasted and that the individual code can be optimally fitted into the FPGA device under individual path delay requirements.

Drawback is that the USB and LabView™ Interfaces were not supported for such individual firmware and hardware solutions and have to be programmed individually by the user: E.g. the steering of the discriminator settings of the SU703 discriminator have to be accomplished by a self developed FPGA user code invoking the serial threshold DAC of this submodule, etc. No further help will be given other than on which pins the SDA/SCL reside on the FPGA side and that the DAC is a 12 bit MAX537 device with 1:1 output ratio to program threshold levels – according to further data given by its datasheet.

All datasheets needed for submodules can be found in the web or the W-Ie-Ne-R file archive.

Hence one has to bring serious professions in VHDL/Verilog coding design to realize the own ideas and to exploit all the hardware features given within the NIMbox/NEMbox.

1.2 Important Comments

The NIMbox/NEMbox is normally shipped with preinstalled firmware which is consistent with the hardware version and with the position of the installed submodules on the main FPGA board.

This paper gives all the printed lists of pin-assignments needed in order to self-construct the .ucf pin assignment files. Additionally .ucf example files can be found in the W-Ie-Ne-R file archive. Further recommendations will be given in this document.

How self made firmware can be downloaded via JTAG or USB will be described in chapter 2.

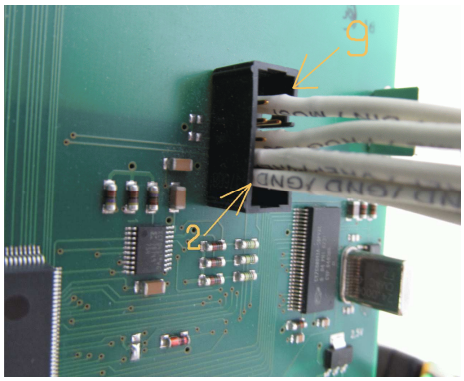
From the first hardware/firmware installation of a NIMbox/NEMbox the user is self responsible for avoiding any damage to the product due to hardware handling and firmware programming activities. E.g. to take serious ESD-precautions and check for I/O directions before connecting.

2 Download of Firmware Files, JTAG and USB Interface

2.1 JTAG Interface Download

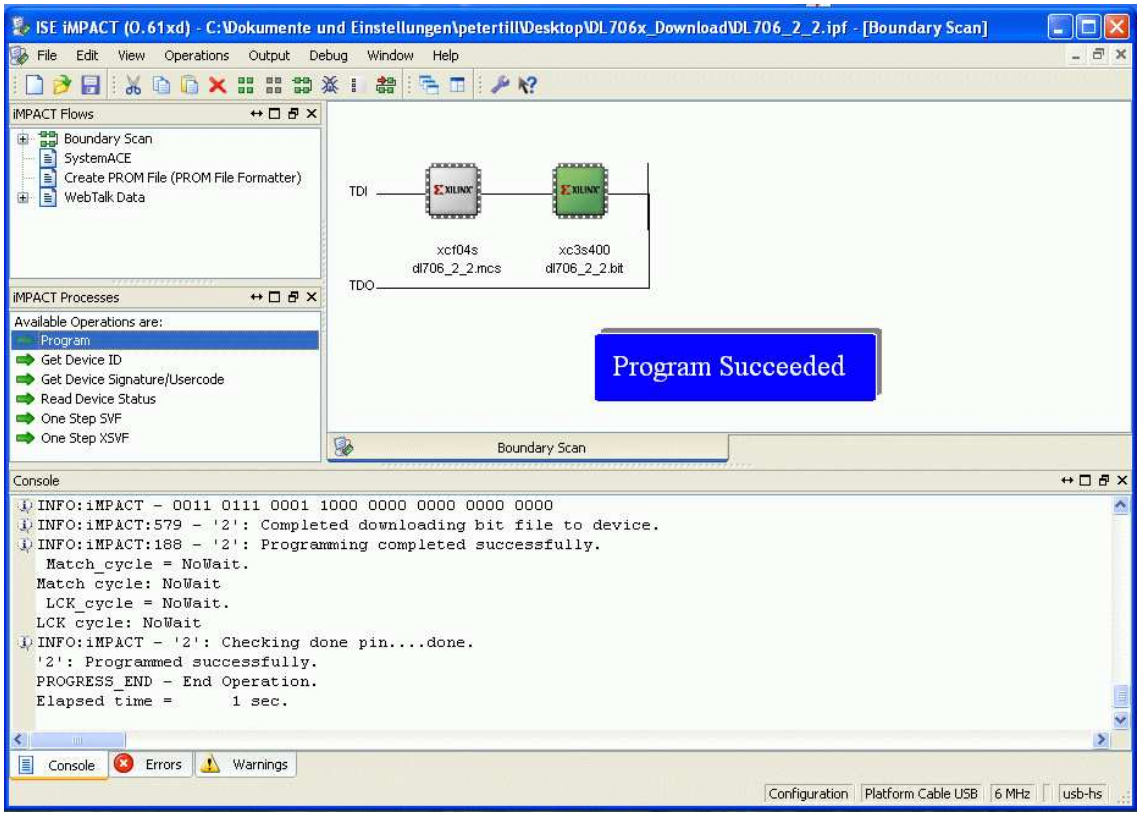
During the development phase fastest and easiest way to download the FPGA firmware is via the JTAG interface. The DL706 mainboard connector contains all pins needed to connect it with the XILINX Flying Wire Adaptor:

Pin 1	TCK	yellow
Pin 2	GND	black
Pin 3	TDO	magenta
Pin 4	VREF	red
Pin 5	TMS	green
Pin 9	TDI	white



The Xilinx product number for the flying wire set is HW-USB-FLYLEADS-G.

Provided that the customer creates FPGA firmware professionally the use of the XILINX Download Tools like IMPACT is not discussed here in detail (manuals can also be found in the file archive). For this case after power-on the JTAG chain shall initialize in a way like this:



Download time for the XC3S400 are seconds and for the XCF04S EEPROM about ten seconds.

Also for use of the other XILINX debugging tools the JTAG interface should work.

2.2 WINDOWS Driver for USB FX2

Once NIMbox/NEMbox has been powered and connected to a USB port, the hardware installation wizard will guide the user through the installation steps. It is recommended not to let Windows look for a proper driver. Choose instead to manually install the driver software from the NIMbox/NEMbox package CD or from the CD repository on the W-Ie-Ne-R file archive.

If the package drive is D, the driver directory is

`D:\(` NIMbox/NEMbox `Version)\USB-Driver-FX2\`

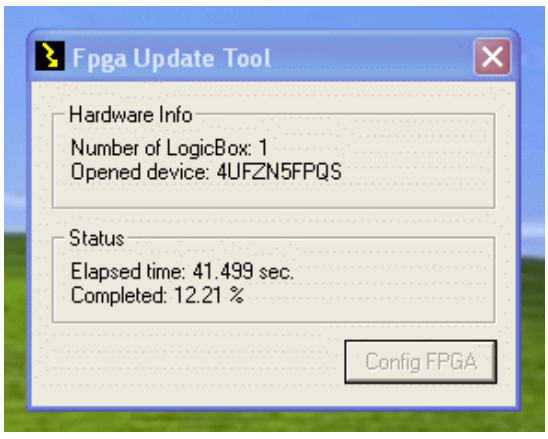
You may look for the driver files with the Hardware Manager of Windows or run the installation script in the above directory.

After successful installation the user can verify the proper operation of NIMbox/NEMbox in the Control Panel: in the Device Manager, there should be a new entry named DL7XX LogicBox.

With this driver it is possible not to use only the USB interface with the NIMbox/NEMbox for I/O data transfer but also for FPGA and EEPROM configuration download.

2.3 Firmware download via USB

To ease firmware download from logistical and mechanical aspects the USB link is the not the fastest but most sophisticated way. To download the corresponding FPGA-.svf-file you simply have to start the "FPGA_Update.exe" program:



This program can be found in the W-Ie-Ne-R file archive in the folder for NIM/NEMbox software. There also can be found some manuals how to generate .svf files out of .bit files. Generally such file generations can be added by additional command statements at the end of a FPGA synthesis process flow to let it generate automatically after every design process.

The download itself lasts about 1-2 minutes.

3 Hardware description

3.1 General description of NIMbox/NEMbox

NIMbox/NEMbox is a programmable module based on a FPGA board (DL706) with 4 slots for I/O submodules that serve as interface between the FPGA I/O signals and the signals in the external environment. It is equipped with a USB port for programming and read out and a connector for direct FPGA programming/debugging.

Its 100 MHz clock makes NIMbox/NEMbox well suited for processing signals with length down to 10 ns and frequencies of several MHz. Such signals are common in nuclear and particle physics applications, where NIM and TTL standards are used for signal transmission and processing.

NIMbox can be used for several application. Typical examples are ADC, discriminator, DAC, NIM/TTL logic, trigger implementations, gate generation and data acquisition.

3.2 Hardware configuration

The main FPGA of the DL706 is a XILINX Spartan-3 Device XC3S400 in PQG208 Package and Speed Grade 5 (Speed Grade 4 is the slower alternative). For this smaller device within the Spartan-3 device family the I/O path delays are much lower than for the big size variants. Please refer to the Spartan-3 Device Family Datasheets for more detailed information.

Solder jumpers around the FPGA are by default set to switch the banks SU<0> – SU<3> to a maximum input swing of 3.465V (default: 3V3, bottom; option: 2V5, top), see fig. 1:

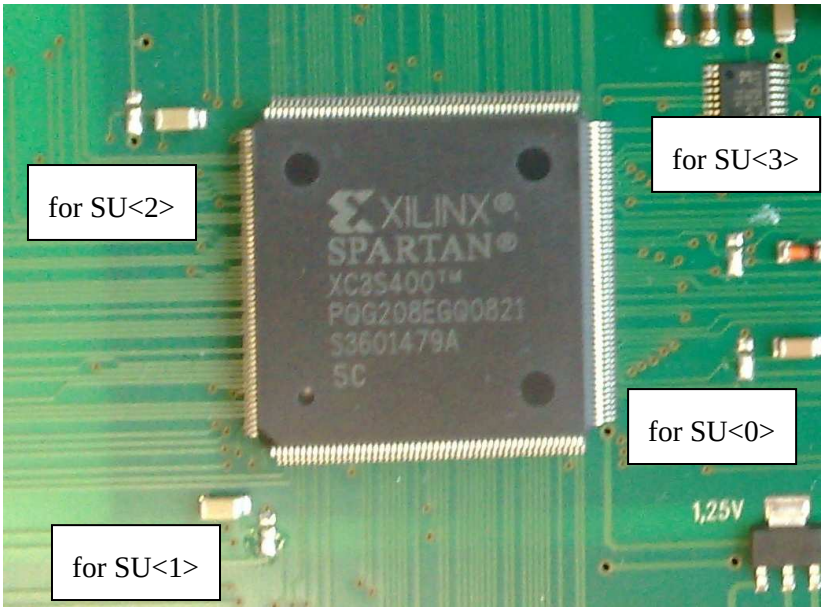


Figure 1: *TheXILINX Spartan-3 Device XC3S400 and I/O Jumper Settings for slots SU<N>*

The slots for submodules on the DL706 mainboard are SU<0>, SU<1>, SU<2> and SU<3> counting upwards from bottom to top. This is the naming convention within this document.

3.3 FPGA I/O Pin Assignment List UCF File

The following is the printout of the standard or basic DL706 user constraint file (.ucf):

```
# DL706 Module Layout Top View  SU<0> SU<1> SU<2> SU<3>
# Start of I/O Pin Assignments
#System
NET "CLK"         LOC = "P181" | IOSTANDARD = LVTTL ;
NET "RES_n"       LOC = "P80"  | IOSTANDARD = LVTTL | PULLUP ;
NET "LED_Back"    LOC = "P184" | IOSTANDARD = LVTTL ;
#USB
NET "FXClk"       LOC = "P77"  | IOSTANDARD = LVTTL ;
NET "FXAddr<0>"   LOC = "P96"  | IOSTANDARD = LVTTL ;
NET "FXAddr<1>"   LOC = "P154" | IOSTANDARD = LVTTL ;
NET "FXData<0>"   LOC = "P92"  | IOSTANDARD = LVTTL ;
NET "FXData<1>"   LOC = "P90"  | IOSTANDARD = LVTTL ;
NET "FXData<2>"   LOC = "P87"  | IOSTANDARD = LVTTL ;
NET "FXData<3>"   LOC = "P86"  | IOSTANDARD = LVTTL ;
NET "FXData<4>"   LOC = "P74"  | IOSTANDARD = LVTTL ;
NET "FXData<5>"   LOC = "P72"  | IOSTANDARD = LVTTL ;
NET "FXData<6>"   LOC = "P68"  | IOSTANDARD = LVTTL ;
NET "FXData<7>"   LOC = "P67"  | IOSTANDARD = LVTTL ;
NET "FXRD_n"      LOC = "P81"  | IOSTANDARD = LVTTL ;
NET "FXWR_n"      LOC = "P83"  | IOSTANDARD = LVTTL ;
NET "FXSLOE_n"    LOC = "P50"  | IOSTANDARD = LVTTL ;
NET "FXEmpty"     LOC = "P58"  | IOSTANDARD = LVTTL ;
NET "FXFull"      LOC = "P57"  | IOSTANDARD = LVTTL ;
NET "FXPEnd_n"    LOC = "P200" | IOSTANDARD = LVTTL ;

#SU<0>
NET "SU<0><3>"    LOC = "P94"  | IOSTANDARD = LVTTL ;
NET "SU<0><4>"    LOC = "P95"  | IOSTANDARD = LVTTL ;
NET "SU<0><5>"    LOC = "P111" | IOSTANDARD = LVTTL ;
NET "SU<0><6>"    LOC = "P113" | IOSTANDARD = LVTTL ;
NET "SU<0><7>"    LOC = "P114" | IOSTANDARD = LVTTL ;
NET "SU<0><8>"    LOC = "P115" | IOSTANDARD = LVTTL ;
NET "SU<0><9>"    LOC = "P116" | IOSTANDARD = LVTTL ;
NET "SU<0><10>"   LOC = "P117" | IOSTANDARD = LVTTL ;
NET "SU<0><11>"   LOC = "P119" | IOSTANDARD = LVTTL ;
NET "SU<0><12>"   LOC = "P120" | IOSTANDARD = LVTTL ;
NET "SU<0><13>"   LOC = "P100" | IOSTANDARD = LVTTL ;
NET "SU<0><14>"   LOC = "P101" | IOSTANDARD = LVTTL ;
NET "SU<0><15>"   LOC = "P106" | IOSTANDARD = LVTTL ;
NET "SU<0><16>"   LOC = "P107" | IOSTANDARD = LVTTL ;
NET "SU<0><17>"   LOC = "P130" | IOSTANDARD = LVTTL ;
NET "SU<0><18>"   LOC = "P131" | IOSTANDARD = LVTTL ;
NET "SU<0><19>"   LOC = "P122" | IOSTANDARD = LVTTL ;
NET "SU<0><20>"   LOC = "P123" | IOSTANDARD = LVTTL ;
NET "SU<0><21>"   LOC = "P138" | IOSTANDARD = LVTTL ;
NET "SU<0><22>"   LOC = "P139" | IOSTANDARD = LVTTL ;
NET "SU<0><23>"   LOC = "P124" | IOSTANDARD = LVTTL ;
NET "SU<0><24>"   LOC = "P125" | IOSTANDARD = LVTTL ;
NET "SU<0><25>"   LOC = "P79"  | IOSTANDARD = LVTTL ;
NET "SU<0><26>"   LOC = "P102" | IOSTANDARD = LVTTL ;
NET "SU<0><27>"   LOC = "P85"  | IOSTANDARD = LVTTL ;
NET "SU<0><28>"   LOC = "P133" | IOSTANDARD = LVTTL ;
NET "SU<0><29>"   LOC = "P93"  | IOSTANDARD = LVTTL ;
NET "SU<0><30>"   LOC = "P126" | IOSTANDARD = LVTTL ;
NET "SU<0><31>"   LOC = "P109" | IOSTANDARD = LVTTL ;
NET "SU<0><32>"   LOC = "P108" | IOSTANDARD = LVTTL ;
```

```
#SU<1>
NET "SU<1><3>"   LOC = "P168" | IOSTANDARD = LVTTL ;
NET "SU<1><4>"   LOC = "P169" | IOSTANDARD = LVTTL ;
NET "SU<1><5>"   LOC = "P150" | IOSTANDARD = LVTTL ;
NET "SU<1><6>"   LOC = "P152" | IOSTANDARD = LVTTL ;
NET "SU<1><7>"   LOC = "P148" | IOSTANDARD = LVTTL ;
NET "SU<1><8>"   LOC = "P149" | IOSTANDARD = LVTTL ;
NET "SU<1><9>"   LOC = "P146" | IOSTANDARD = LVTTL ;
NET "SU<1><10>"  LOC = "P147" | IOSTANDARD = LVTTL ;
NET "SU<1><11>"  LOC = "P143" | IOSTANDARD = LVTTL ;
NET "SU<1><12>"  LOC = "P144" | IOSTANDARD = LVTTL ;
NET "SU<1><13>"  LOC = "P155" | IOSTANDARD = LVTTL ;
NET "SU<1><14>"  LOC = "P156" | IOSTANDARD = LVTTL ;
NET "SU<1><15>"  LOC = "P161" | IOSTANDARD = LVTTL ;
NET "SU<1><16>"  LOC = "P162" | IOSTANDARD = LVTTL ;
NET "SU<1><17>"  LOC = "P176" | IOSTANDARD = LVTTL ;
NET "SU<1><18>"  LOC = "P178" | IOSTANDARD = LVTTL ;
NET "SU<1><19>"  LOC = "P140" | IOSTANDARD = LVTTL ;
NET "SU<1><20>"  LOC = "P141" | IOSTANDARD = LVTTL ;
NET "SU<1><21>"  LOC = "P165" | IOSTANDARD = LVTTL ;
NET "SU<1><22>"  LOC = "P166" | IOSTANDARD = LVTTL ;
NET "SU<1><23>"  LOC = "P171" | IOSTANDARD = LVTTL ;
NET "SU<1><24>"  LOC = "P172" | IOSTANDARD = LVTTL ;
NET "SU<1><25>"  LOC = "P180" | IOSTANDARD = LVTTL ;
NET "SU<1><26>"  LOC = "P132" | IOSTANDARD = LVTTL ;
NET "SU<1><27>"  LOC = "P175" | IOSTANDARD = LVTTL ;
NET "SU<1><28>"  LOC = "P182" | IOSTANDARD = LVTTL ;
NET "SU<1><29>"  LOC = "P167" | IOSTANDARD = LVTTL ;
NET "SU<1><30>"  LOC = "P135" | IOSTANDARD = LVTTL ;
NET "SU<1><31>"  LOC = "P128" | IOSTANDARD = LVTTL ;
NET "SU<1><32>"  LOC = "P137" | IOSTANDARD = LVTTL ;

#SU<2>
NET "SU<2><3>"   LOC = "P190" | IOSTANDARD = LVTTL ;
NET "SU<2><4>"   LOC = "P191" | IOSTANDARD = LVTTL ;
NET "SU<2><5>"   LOC = "P194" | IOSTANDARD = LVTTL ;
NET "SU<2><6>"   LOC = "P196" | IOSTANDARD = LVTTL ;
NET "SU<2><7>"   LOC = "P198" | IOSTANDARD = LVTTL ;
NET "SU<2><8>"   LOC = "P199" | IOSTANDARD = LVTTL ;
NET "SU<2><9>"   LOC = "P10"  | IOSTANDARD = LVTTL ;
NET "SU<2><10>"  LOC = "P11"  | IOSTANDARD = LVTTL ;
NET "SU<2><11>"  LOC = "P12"  | IOSTANDARD = LVTTL ;
NET "SU<2><12>"  LOC = "P13"  | IOSTANDARD = LVTTL ;
NET "SU<2><13>"  LOC = "P2"   | IOSTANDARD = LVTTL ;
NET "SU<2><14>"  LOC = "P3"   | IOSTANDARD = LVTTL ;
NET "SU<2><15>"  LOC = "P203 "| IOSTANDARD = LVTTL ;
NET "SU<2><16>"  LOC = "P204" | IOSTANDARD = LVTTL ;
NET "SU<2><17>"  LOC = "P26"  | IOSTANDARD = LVTTL ;
NET "SU<2><18>"  LOC = "P27"  | IOSTANDARD = LVTTL ;
NET "SU<2><19>"  LOC = "P185" | IOSTANDARD = LVTTL ;
NET "SU<2><20>"  LOC = "P187" | IOSTANDARD = LVTTL ;
NET "SU<2><21>"  LOC = "P15"  | IOSTANDARD = LVTTL ;
NET "SU<2><22>"  LOC = "P16"  | IOSTANDARD = LVTTL ;
NET "SU<2><23>"  LOC = "P7"   | IOSTANDARD = LVTTL ;
NET "SU<2><24>"  LOC = "P9"   | IOSTANDARD = LVTTL ;
NET "SU<2><25>"  LOC = "P183" | IOSTANDARD = LVTTL ;
NET "SU<2><26>"  LOC = "P205" | IOSTANDARD = LVTTL ;
NET "SU<2><27>"  LOC = "P189" | IOSTANDARD = LVTTL ;
NET "SU<2><28>"  LOC = "P197" | IOSTANDARD = LVTTL ;
NET "SU<2><29>"  LOC = "P29"  | IOSTANDARD = LVTTL ;
NET "SU<2><30>"  LOC = "P22"  | IOSTANDARD = LVTTL ;
NET "SU<2><31>"  LOC = "P5"   | IOSTANDARD = LVTTL ;
NET "SU<2><32>"  LOC = "P4"   | IOSTANDARD = LVTTL ;
```

```
#SU<3>
NET "SU<3><3>"   LOC = "P46" | IOSTANDARD = LVTTL ;
NET "SU<3><4>"   LOC = "P48" | IOSTANDARD = LVTTL ;
NET "SU<3><5>"   LOC = "P44" | IOSTANDARD = LVTTL ;
NET "SU<3><6>"   LOC = "P45" | IOSTANDARD = LVTTL ;
NET "SU<3><7>"   LOC = "P42" | IOSTANDARD = LVTTL ;
NET "SU<3><8>"   LOC = "P43" | IOSTANDARD = LVTTL ;
NET "SU<3><9>"   LOC = "P39" | IOSTANDARD = LVTTL ;
NET "SU<3><10>"  LOC = "P40" | IOSTANDARD = LVTTL ;
NET "SU<3><11>"  LOC = "P36" | IOSTANDARD = LVTTL ;
NET "SU<3><12>"  LOC = "P37" | IOSTANDARD = LVTTL ;
NET "SU<3><13>"  LOC = "P51" | IOSTANDARD = LVTTL ;
NET "SU<3><14>"  LOC = "P52" | IOSTANDARD = LVTTL ;
NET "SU<3><15>"  LOC = "P61" | IOSTANDARD = LVTTL ;
NET "SU<3><16>"  LOC = "P62" | IOSTANDARD = LVTTL ;
NET "SU<3><17>"  LOC = "P64" | IOSTANDARD = LVTTL ;
NET "SU<3><18>"  LOC = "P65" | IOSTANDARD = LVTTL ;
NET "SU<3><19>"  LOC = "P34" | IOSTANDARD = LVTTL ;
NET "SU<3><20>"  LOC = "P35" | IOSTANDARD = LVTTL ;
NET "SU<3><21>"  LOC = "P18" | IOSTANDARD = LVTTL ;
NET "SU<3><22>"  LOC = "P19" | IOSTANDARD = LVTTL ;
NET "SU<3><23>"  LOC = "P20" | IOSTANDARD = LVTTL ;
NET "SU<3><24>"  LOC = "P21" | IOSTANDARD = LVTTL ;
NET "SU<3><25>"  LOC = "P76" | IOSTANDARD = LVTTL ;
NET "SU<3><26>"  LOC = "P28" | IOSTANDARD = LVTTL ;
NET "SU<3><27>"  LOC = "P78" | IOSTANDARD = LVTTL ;
NET "SU<3><28>"  LOC = "P71" | IOSTANDARD = LVTTL ;
NET "SU<3><29>"  LOC = "P63" | IOSTANDARD = LVTTL ;
NET "SU<3><30>"  LOC = "P24" | IOSTANDARD = LVTTL ;
NET "SU<3><31>"  LOC = "P31" | IOSTANDARD = LVTTL ;
NET "SU<3><32>"  LOC = "P33" | IOSTANDARD = LVTTL ;
# Initialize
INST "Cntrl/USBCnt_0" INIT = 1;
# Timing
#---------------------------NIM/NEMbox-------------------------
# Timing
NET "CLK" TNM_NET = "CLK";
TIMESPEC "TS_CLK" = PERIOD "CLK" 10 ns HIGH 50 %;
#
INST "Ticker20ms*" TNM = "Tick1ms";
TIMESPEC "TS_Cnt1ms" = FROM "Tick1ms" TO "Tick1ms" 1 ms;
#
NET "*_TIG*" TIG;
#
```

The first three lines connect the FPGA for 100 MHz clock, RESET button and the BUSY LED.

The second part describes the connections to the common interface of the FPGA to the USB interface (8 bit) and EEPROM (1 bit). The prior is described in more detail in the next section.

As you can see 29 of the 34 pins of the I/O connectors are linked to the FPGA by bidirectional SU<I><J> pins, which have to be initialized all to HIGH-Z with a basic VHDL code.

All signals should be declared in such an UCF file even once, also if not directly used by the VHDL/Verilog code.

3.4 FX2 USB I/O, RESET, LED and CLOCK

The FX2 USB connection is realized with a USB-chip from Cypress CY7C68013-56 attached to a 24 MHz oscillator. The implementation of the fixed installed core will be done with the following VHDL code corresponding to the naming conventions given in the UCF file:

```
entity DL706 is
    Port (  CLK        : in    std_logic;
            RES_n      : in    std_logic;
            LED        : out   std_logic;
            -- FX2
            FXClk      : out   std_logic;
            FXAddr     : out   std_logic_vector(1 downto 0);
            FXData     : inout std_logic_vector(7 downto 0);
            FXRD_n     : out   std_logic;
            FXWR_n     : out   std_logic;
            FXSLOE_n   : inout std_logic;
            FXEmpty    : in    std_logic;
            FXFull     : in    std_logic;
            FXPEnd_n   : inout std_logic;
            -- SU7xx Modules
            SU         : inout SU_Connectors(0 to 3));
end DL706;
```

If you like to use the USB port for own data transmition purposes please install your own driver interface of your choice – means your implementation specific behavioral architecture.

3.5 I/O Connectors

The I/O connectors are standard two row 36 pin 100 mil IDC receptacles. The counting starts from the board top view from the **upper right** (pin 1) to the **lower left** (pin 36) for the NIMbox/NEMbox I/O submodules naming conventions.

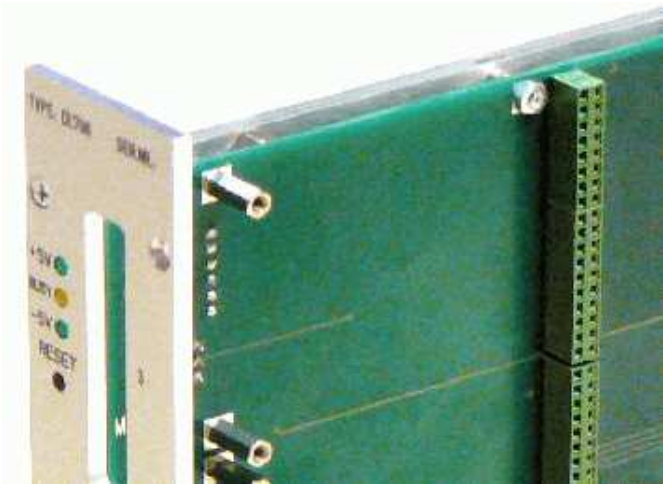


Figure 2: The NIMbox/NEMbox I/O submodule connector (green) position SU<3>

The SU<i><j> coding convention as in the UCF file is for <j> same as the pin number of the IDC connector, e.g. Pin 3 (j=3) of the uppermost module (i=3) is connected to P46 of the FPGA.

3.6 I/O submodules

Presently, table 1 sums up all available submodules for NIMbox/NEMbox. Figure 3 gives the mechanical dimensions with the connector position can be derived. The maximum allowed part height is 7.5 mm on bottom side and 2-3 mm on top side respectively. Table 2 lists the standard pin-out of the I/O submodules. For the accepted I/O levels refer to the Spartan-3 family sheets.

Submodule	Function
SU703	4x Discriminator and 1x TTL I/O – LEMO COAX
SU704	5x NIM/TTL I/O – LEMO COAX
SU706	1x ADC (100 Mhz), 2x TTL I/O – LEMO COAX
SU707	8x LVDS I/O
SU709	8x Temperature Sensor
SU710	2x Fast DAC (100 MHz)
SU711	6x programmable delay line 0,5 ns .. 128 ns

Table 1: NIMbox/NEMbox I/O submodules

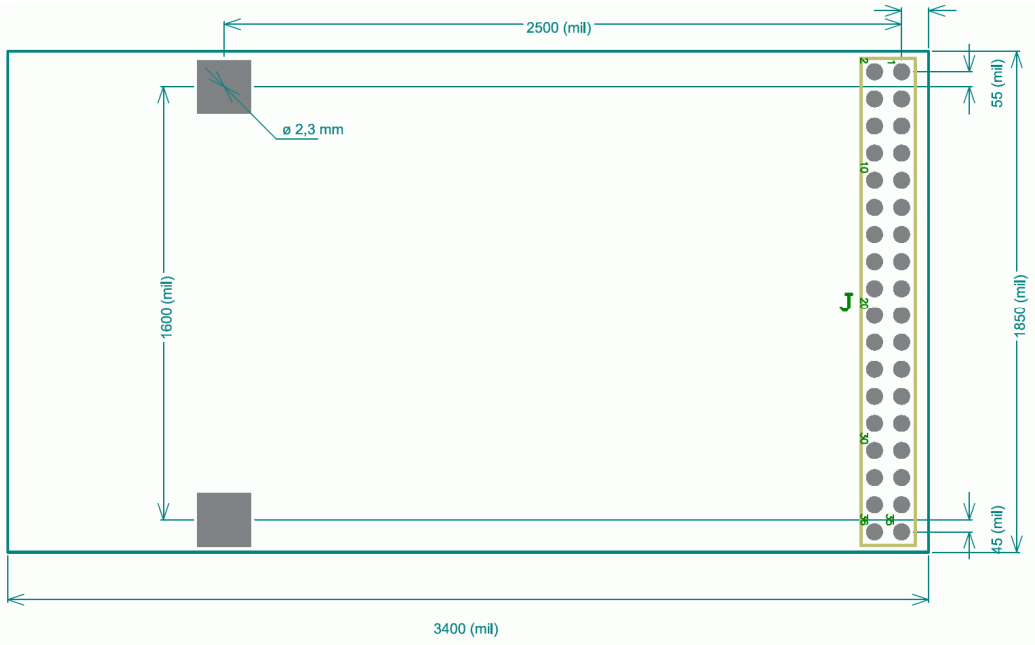


Figure 3: Mechanical card dimensions of a submodule and relative pin-1 position

Pins	Signal
1, 2	+5V power
33	NC (not connected)
34	VCCO (+3V3 or +2V5 jumper selectable)
35, 36	GND power

Table 2: Standard Pin – Outs of all I/O submodules

To generate other voltages like -5V take ultra low profile DC/DC converters of your choice (e.g. NME line from MURATA, PM line from PEAK, TDK, LTC etc.) Use pin 34 only as reference or for very low power consumers (< 10 mA per submodule or < 40 mA in sum).

3.7 SU703 – Discriminator I/O

3.7.1 General Description

SU703 has 1 to 4 discriminator inputs and 4 to 1 TTL I/O ports, where the total sum of the devices is limited to 5, i.e. the number of LEMO COAX connectors on the front. The first LEMO connector corresponds always to a discriminator, while the last one always to a TTL I/O port. Connectors 2 to 4 are discriminators by default but their functionality can be changed with minor hardware modifications.

Discriminators thresholds can be programmed within the –2,5 / +2,5 V range and discriminators hysteresis within the 0 / 60 mV range, both with 12 bit resolution. The propagation delay of the used MAX9602 is below a nanosecond hence the capable speed is limited by the FPGA only.

TTL I/Os can generate a current of more than 60 mA and therefore a sufficient TTL level greater 2.4V with a 50 Ω terminated coaxial line. If used as inputs, TTL I/Os must be terminated with 50 Ω. Alternatively, it is possible to bring the input port high through a pull up resistor of about 1 kΩ, and through a simple switch it is possible to bring it down by shorting.

3.7.2 SU703 (discriminator) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVDS_25_n	in	D_out_n<1>	4	3	D_out_p<1>	in	LVDS_25_p
LVDS_25_n	in	D_out_n<2>	6	5	D_out_p<2>	in	LVDS_25_p
LVDS_25_n	in	D_out_n<3>	8	7	D_out_p<3>	in	LVDS_25_p
LVDS_25_n	in	D_out_n<4>	10	9	D_out_p<4>	in	LVDS_25_p
LVTTL	out	SCLK_DAC	12	11	SDI_DAC	out	LVTTL
LVTTL	out	CS_TH_n	14	13	LDAC_n	out	LVTTL
LVTTL	in	Dout<1>	16	15	CS_HYS_n	out	LVTTL
LVTTL	in	Dout<3>	18	17	Dout<2>	in	LVTTL
LVTTL	out	DOE_n<4>	20	19	Din<4>	out	LVTTL
LVTTL	out	DOE_n<3>	22	21	Din<3>	out	LVTTL
LVTTL	out	DOE_n<2>	24	23	Din<2>	out	LVTTL
LVTTL	out	DOE_n<1>	26	25	Din<1>	out	LVTTL
LVTTL	out	LED_n<5>	28	27	LED_n<4>	out	LVTTL
LVTTL	out	LED_n<3>	30	29	LED_n<2>	out	LVTTL
LVTTL	in	Dout<4>	32	31	LED_n<1>	out	LVTTL

So, in the UCF file you have to alter all LVDS differential lines to the LVDS standard, e.g.:

```
NET "SU<0><3>"  LOC = "P94" | IOSTANDARD = LVDS_25 ;
NET "SU<0><4>"  LOC = "P95" | IOSTANDARD = LVDS_25 ;
```

The LVDS pairing is the done the same way for all banks as it is done in the list above.

Threshold and hysteresis DAC are both 12 bit MAX537s. Please refer to the datasheet for programming via the SDI/SCLK_DAC lines.

Discriminators and TTL Inputs are working in parallel and without interference, as long as the TTL output drivers were not output enabled. Keep the negative logic output enable lines DOE_n<I> HIGH as long as no other source is adapted to the LEMO receptacle. Else one can destroy the (zero ohms output fuses in front of the) TTL drivers

3.8 SU704 – NIM I/O

3.8.1 General Description

SU704 has 5 identical LEMO COAX I/O connectors to be used as programmable digital I/O ports. Every I/O port supports both NIM and TTL levels, but the selected level is defined by a jumper setting on the SU704 board (default level is NIM). Input impedance can be set to 50 Ω through a relays.

Every NIM output can draw a current of -16 mA, thus with a 50 Ω impedance the level is -0,8V. The corresponding input threshold, with 50 Ω set, is -0,4V.

Maximum frequency of the submodule itself is about 200 MHz and propagation delay 3 ns. Hence path delay and maximum toggle frequency are mainly given by the FPGA.

3.8.2 SU704 (NIM I/O) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	out	OUTP<5>	4	3	INTTL<5>	in	LVTTL
LVTTL	out	OE_n<5>	6	5	INNIM<5>	in	LVTTL
LVTTL	HIGH-Z or in	-	8	7	LED_n<5>	out	LVTTL
LVTTL	out	OUTP<4>	10	9	INTTL<4>	in	LVTTL
LVTTL	out	OE_n<4>	12	11	INNIM<4>	in	LVTTL
LVTTL	HIGH-Z or in	-	14	13	LED_n<4>	out	LVTTL
LVTTL	out	OUTP<3>	16	15	INTTL<3>	in	LVTTL
LVTTL	out	OE_n<3>	18	17	INNIM<3>	in	LVTTL
LVTTL	HIGH-Z or in	-	20	19	LED_n<3>	out	LVTTL
LVTTL	out	OUTP<2>	22	21	INTTL<2>	in	LVTTL
LVTTL	out	OE_n<2>	24	23	INNIM<2>	in	LVTTL
LVTTL	HIGH-Z or in	-	26	25	LED_n<2>	out	LVTTL
LVTTL	out	OUTP<1>	28	27	INTTL<1>	in	LVTTL
LVTTL	out	OE_n<1>	30	29	INNIM<1>	in	LVTTL
LVTTL	out	-	32	31	LED_n<1>	out	LVTTL

3.9 SU706 – Fast ADC submodule

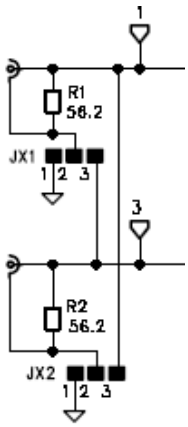
3.9.1 General Description

The two upper LEMO COAX receptacles ADC+/ADC- are connected to the fast differential (Flash-) ADC (ADS5500). The ADC is of 14 bit resolution with 100 MHz sampling rate.

The input is coupled passively with a differential transformer of 20 kHz to 50 MHz frequency range and the input voltage range is within 0,5..2,3 Vpp depending on the transformer ratio (1:16 .. 1:1), default range is -0,5 V to +0,5 V.

With an external 0 Ω LEMO COAX terminator adapted to one pole of the upper two LEMO COAX receptacles the ADC mode can be switched to single positive or negative polarity.

The grounding of the two ADC receptacles can be isolated from case ground by opening of additional solder jumpers. The default jumper settings in the input circuitry behind the ADC connectors (left) is as follows:



By default JX1 and JX2 are set to 1-2.

The lower two LEMO COAX receptacles T<N>/T<N+1> are connected to TTL-I/O-drivers each. Normally they will be used as CLOCK and TRIGGER input, but if those signals come from the FPGA itself those drivers can be used for alternative things.

3.9.2 SU706 (Fast ADC) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	out	RESET	4	3	CLKin	in	LVTTL
LVTTL	in	DATA<0>	6	5	DATA<1>	in	LVTTL
LVTTL	in	DATA<2>	8	7	DATA<3>	in	LVTTL
LVTTL	in	DATA<4>	10	9	DATA<5>	in	LVTTL
LVTTL	in	DATA<6>	12	11	DATA<7>	in	LVTTL
LVTTL	in	DATA<8>	14	13	DATA<9>	in	LVTTL
LVTTL	in	DATA<10>	16	15	DATA<11>	in	LVTTL
LVTTL	in	DATA<12>	18	17	DATA<13>	in	LVTTL
LVTTL	in	OVR	20	19	SCLKn	out	LVTTL
LVTTL	out	SDATA	22	21	SEn	out	LVTTL
LVTTL	out	CLKn	24	23	CLKp	out	LVTTL
LVTTL	out	DIO2OUT	26	25	DIO2OEn	out	LVTTL
LVTTL	out	DIO1OEn	28	27	DIO2IN	in	LVTTL
LVTTL	out	DIO1OUT	30	29	DIO1IN	in	LVTTL
LVTTL	out	LED1n	32	31	LED2n	out	LVTTL

Please refer to the ADS5500 datasheet for functions.

3.10 SU707 – LVDS I/O submodule

3.10.1 General Description

The module contains 8 LVDS-I/Os which are separated into two groups of RJ45 receptacles.

Every group (1..2) has 4 channels (A..D) and can be programmed within those groups whether as inputs or outputs.

3.10.2 SU707 (LVDS IO) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	in	LVDS_I<1>	4	3	LVDS_I<2>	in	LVTTL
LVTTL	out	LVDS_O<1>	6	5	LVDS_O<2>	out	LVTTL
LVTTL	out	LVDS_O<3>	8	7	LVDS_O<4>	out	LVTTL
LVTTL	in	LVDS_I<3>	10	9	LVDS_I<4>	in	LVTTL
LVTTL	in	LVDS_I<5>	12	11	LVDS_I<6>	in	LVTTL
LVTTL	out	LVDS_O<5>	14	13	LVDS_O<6>	out	LVTTL
LVTTL	out	LVDS_O<7>	16	15	LVDS_O<8>	out	LVTTL
LVTTL	in	LVDS_I<7>	18	17	LVDS_I<8>	in	LVTTL
LVTTL	out	EN_O<2>	20	19	EN_I<2>	out	LVTTL
LVTTL	out	EN_O<1>	22	21	EN_I<1>	out	LVTTL
LVTTL	HIGH-Z or in	-	24	23	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	26	25	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	28	27	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	30	29	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	32	31	-	HIGH-Z or in	LVTTL

The conversion is done via SN75LVDS391PW transceiver chips so that all levels are LVTLL. Refer to the datasheet for I/O directional checks – EN signals are directly passed to the FPGA.

3.11 SU709 – Temperature Sensors submodule

3.11.1 General Description

This submodule serves as interface to the SMT16030 (SMARTEC) temperature sensors which contain integrated analogue to frequency converters. There the temperature is transformed into a calibrated on chip proportional duty cycle (DS) put to a digital output.

The SU709 has 8 RM2.54 mm DUBOX connectors with GND, DS, +5V each for direct supply.

Default range of the sensor is -45° to +130°C.

3.11.2 SU709 (Temperature Sensors) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	in	TSEN<1>	4	3	TSEN<0>	in	LVTTL
LVTTL	in	TSEN<3>	6	5	TSEN<2>	in	LVTTL
LVTTL	in	TSEN<5>	8	7	TSEN<4>	in	LVTTL
LVTTL	in	TSEN<7>	10	9	TSEN<6>	in	LVTTL
LVTTL	HIGH-Z or in	-	12	11	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	14	13	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	16	15	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	18	17	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	20	19	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	22	21	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	24	23	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	26	25	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	28	27	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	30	29	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	32	31	-	HIGH-Z or in	LVTTL
LVTTL	HIGH-Z or in	-	34	33	-	HIGH-Z or in	LVTTL

Please refer to the SMT16030 types of datasheets for the conversion formulae.

3.12 SU710 – Fast DAC submodule

3.12.1 General Description

This module is a fast 2 channel digital to analogue converter (DAC2904) with 14 bit resolution each. Both channels (A & B) are completely independent and run with a sampling rate of up to 125 MHz. Typical applications are for the generation of complex waveforms (AFG = Arbitrary Waveform Generator).

The modules circuitry allows for single operation with -2V .. +2V range or dual line operation with -1V .. +1V range. This behavior result of a to 50 Ω optimized current amplifier for the first LEMO COAX output with the same current mirrored to the second output respectively.

3.12.2 SU710 (Fast DAC) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	out	DATA1<0>	4	3	DATA1<1>	out	LVTTL
LVTTL	out	DATA1<2>	6	5	DATA1<3>	out	LVTTL
LVTTL	out	DATA1<4>	8	7	DATA1<5>	out	LVTTL
LVTTL	out	DATA1<6>	10	9	DATA1<7>	out	LVTTL
LVTTL	out	DATA1<8>	12	11	DATA1<9>	out	LVTTL
LVTTL	out	DATA1<10>	14	13	DATA1<11>	out	LVTTL
LVTTL	out	DATA1<12>	16	15	DATA1<13>	out	LVTTL
LVTTL	out	LOAD2	18	17	LOAD1	out	LVTTL
LVTTL	out	DATA2<0>	20	19	DATA2<1>	out	LVTTL
LVTTL	out	DATA2<2>	22	21	DATA2<3>	out	LVTTL
LVTTL	out	DATA2<4>	24	23	DATA2<5>	out	LVTTL
LVTTL	out	DATA2<6>	26	25	DATA2<7>	out	LVTTL
LVTTL	out	DATA2<8>	28	27	DATA2<9>	out	LVTTL
LVTTL	out	DATA2<10>	30	29	DATA2<11>	out	LVTTL
LVTTL	out	DATA2<12>	32	31	DATA2<13>	out	LVTTL

The programming is relatively easy – please refer to the DAC2904 sheet.

3.13 SU711 – 5 Channel Delay 0.5 to 128 ns

3.13.1 General Description

This module is capable to delay digital signals for up to 5 channels asynchronously – means independent of any clock – by programmable amounts of time in steps of a half nanosecond and with up to 128 steps. The corresponding delay chip DS1023-50 has a basic delay of about 16,5 nanoseconds..

3.13.2 SU710 (Fast DAC) pin assignment

Pin numbers are in pale green as from top view to connector. Left half gives information for the left row, right half for the right row. Pale yellow are the corresponding levels. Pale red are the I/O directions from view of the FPGA. Pins 1, 2, 33-36 are not connected to the FPGA.

LVTTL	out	DATA<1>	4	3	DATA<0>	out	LVTTL
LVTTL	out	DATA<3>	6	5	DATA<2>	out	LVTTL
LVTTL	out	DATA<5>	8	7	DATA<4>	out	LVTTL
LVTTL	out	DATA<7>	10	9	DATA<6>	out	LVTTL
LVTTL	out	LE<1>	12	11	DELout<1>	out	LVTTL
LVTTL	in	REF<1>	14	13	DELin<1>	in	LVTTL
LVTTL	out	LE<2>	16	15	DELout<2>	out	LVTTL
LVTTL	in	REF<2>	18	17	DELin<2>	in	LVTTL
LVTTL	out	LE<3>	20	19	DELout<3>	out	LVTTL
LVTTL	in	REF<3>	22	21	DELin<3>	in	LVTTL
LVTTL	out	LE<4>	24	23	DELout<4>	out	LVTTL
LVTTL	in	REF<4>	26	25	DELin<4>	in	LVTTL
LVTTL	out	LE<5>	28	27	DELout<5>	out	LVTTL
LVTTL	in	REF<5>	30	29	DELin<5>	in	LVTTL
LVTTL	HIGH-Z or in	-	32	31	-	HIGH-Z or in	LVTTL

Please refer to the DS1023-50 data sheet for the programming structure and the special meaning of signals